Energy Isolation Required for Multi-tenant Energy Harvesting Platforms

Branden Ghena

Joshua Adkins, Bradford Campbell, Branden Ghena, Neal Jackson, Pat Pannuto, & Prabal Dutta

ENSsys'17
November 5, 2017 - Delft, Netherlands

It’s time to start thinking about energy-harvesting and multi-tenancy together

- How do we support multiple unaligned applications while also balancing limited energy availability?

- We need to address energy sharing, isolation, and adaptivity

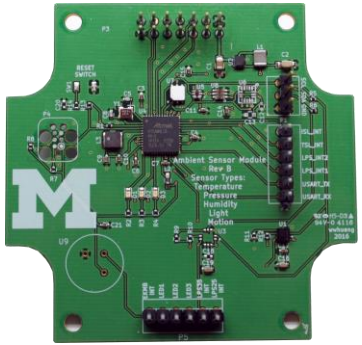
Motivation: The Signpost Platform

- Enabling city-scale sensing applications
  - Platform provides base services
  - Modules run applications and include sensors

- Infrastructure-free infrastructure
  - Solar energy harvesting
  - Multiple wireless networking options
  - Easy (two bolt) installation

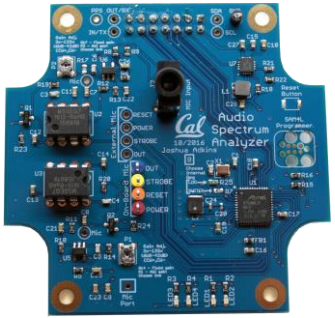
Signpost expects a variety of applications

RF Spectrum Use Measurement



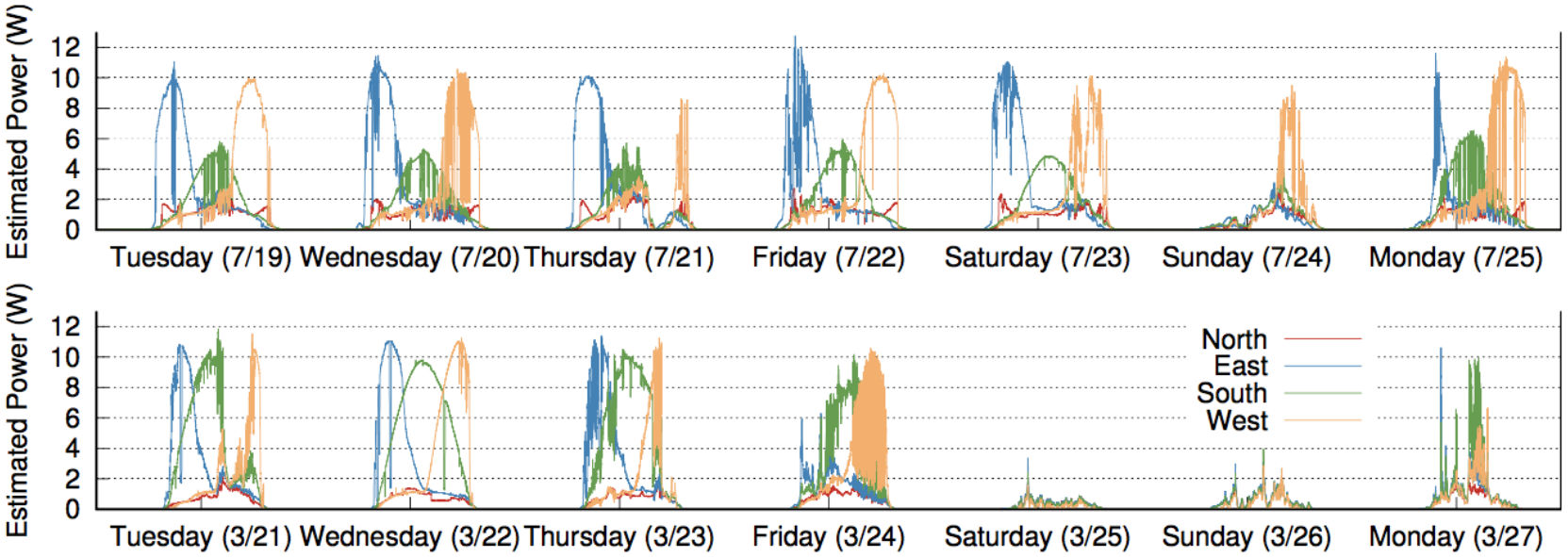
Environmental Sensing

Motion Detection



Audio Analysis

Signpost expects energy variability



- Daily average varies between 2.7 W and 0.09 W

Design requirements for Signpost

- Energy on Signpost is shared between applications
  - Needs individual module energy use measurements
  - Needs individual module power control
  - Needs to measure and charge for service usage
  - Needs to allocate incoming energy among applications

- Platform needs to enable adaptable applications
  - Expects to be deployed with varying energy availability
  - Applications can’t be re-written for each Signpost

Multiprogramming is coming to single-microcontroller systems too

- Systems like Tock and Amulet provide multi-tenancy on low-power, resource-constrained systems

- These systems bring new challenges
  - Applications may be even further decoupled from the hardware
  - Fine-grained measurements of energy use may be difficult
  - The platform needs to account for and be able to reset external state

How do we enable application reasoning?

- Guarantee: allocated energy may only decrease predictably
  - The application may run and use the energy
  - Services may be used that charge against the application
  - The platform may charge a constant upkeep cost

- Allow applications to plan for the worst-case future

How should energy be allocated to applications?

- Simple: Divide evenly between virtual allocations
- Complex: Use energy allocation as a form of priority

- Design questions
  - Need to decide how much energy each application can store
  - Need to decide how frequently to apportion incoming energy
    - Lack of care leads to intermittent operation

Discussion

How do we make energy-limited programming accessible to developers?

- Allocation guarantees enable reasoning, but only if applications can access and interpret their energy availability
- Do we really want application developers to have to reason about their energy?
- Is run-to-completion at a varying frequency the only model that works here?

Discussion

How do we enable developers to understand how code is functioning at runtime?

- Is compile-time analysis still helpful in a multiprogramming scenario?
- How do you report runtime performance of applications?
- How do you know in advance that a certain combination of applications is going to have undesirable results?

Discussion

When is state-based profiling an accurate enough as a measure of application energy use?

- Putting energy gauges everywhere is unlikely for many platforms
- Accurately sharing a hardware resource, like a radio, is difficult even with an individual gauge

Discussion

What reliability guarantees can we expect from energy-harvesting systems?

- **How do you distinguish failure from lack of energy?**
- **Can we include energy-harvesting systems in user facing applications?**

For More Information…

- https://github.com/lab11
  - lab11/signpost [hardware design files]
  - lab11/signpost-software [software repository]

- https://github.com/helena-project/tock

- Email: brghena@berkeley.edu